



ICONI



KOREAN SOCIETY FOR INTERNET INFORMATION

The 11th International Conference on Internet

(ICONI 2019)

Dec. 15-18, 2019, Lotte Hotels & Resorts Hanoi, Vietnam

<http://www.iconi.org>

Proceedings of ICONI 2019

| Organized by |

Korean Society for Internet Information (KSII)

| Sponsored by |

Korea Electronics Technology Institute (KETI)

Contents

1-1	Mobility-Aware Service Migration (MASM): An Edge Cloud Service Migration Algorithm for Seamless Mobility Haziq Hamzah, Duc-Tai Le (Sungkyunkwan Univ., ROK), Moonseong Kim (Seoul Theological Univ., ROK), Hyunseung Choo (Sungkyunkwan Univ., ROK)	1-3
1-2	Effects of Shoe Heel Types on Knee Joint Angle During Gait in Healthy Young Women – A Preliminary Study Vantha Chhoeum, Changwon Wang, Seungwan Jang, Se Dong Min, Young Kim (Soonchunhyang Univ., ROK), Min-Hyung Choi (Univ. of Colorado at Denver, USA)	4-8
1-3	Transfer Learning: A Practical Approach to Android Malware Classification Long Nguyen-Vu, Souhwan Jung (Soongsil Univ., ROK)	9-14
1-4	Automatic Generation of MAEC Standards for Android Malware Threat Intelligence Jungsoo Park, Nguyen Vu Long (Soongsil Univ., ROK), George Bencivengo (GINIA Inc, Washington DC, USA), Souhwan Jung (Soongsil Univ., ROK)	15-20
1-5	“The Doctor Knows You” : A Telemedicine System to bring medical services to rural and remote villages in Thailand Sirikasem Sirilak, Songsak Thongsanit, Roy Morien (Naresuan Univ., Thailand)	21-26
1-6	Event-based Traffic Monitoring Architecture for Detecting Click-fraud Applications Hae-Seon Jeong, Jin Kwak (Ajou Univ., ROK)	27-30
2-1	A Reviews of a Leader’ s Voice Traits and Members’ Cognitions: The Relationship between Leader’ s Voice and Trust Sangwoo Hahm, Hyungwoo Park (Soongsil Univ., ROK)	31-34
2-2	Understanding the acceptance of subscription-based online services Dong Hyuk Jo (Soongsil Univ., ROK)	35-42
2-3	Unusual Indoor Event Detection System using Vision, Audio, and Activity Sensors Ju Ho Jung, Siwon Kim, Ryumduk Oh (Korea National Univ. of Transportation, ROK), Heungseok Jeon (Konkuk Univ., ROK), Junho Ahn (Korea National Univ. of Transportation, ROK)	43-46

Event-based Traffic Monitoring Architecture for Detecting Click-fraud Applications

Hae-Seon Jeong¹ and Jin Kwak²

¹ISAA Lab., Department of Computer Engineering, Ajou University
Suwon, Republic of Korea

²Department of Cyber Security, Ajou University
Suwon, Republic of Korea

[e-mail: hsjeong.isaa@gmail.com, security@ajou.ac.kr]

*Corresponding author: Jin Kwak

Abstract

A potentially harmful application (PHA) refers to an application that functions normally but poses a potential security risk to users and their data. As of 2018, 54.9% of apps classified as PHAs are click-fraud apps, which continue to send advertisement click signals without the user's knowledge to maximize ad revenue. Abnormal advertising behavior can result in the misuse of user device resources and secondary damage such as user information leakage. Thus, in this paper, an event-based traffic monitoring technique is presented to detect attacks by click-fraud apps.

Keywords: Click-fraud, Android, PHA, Traffic Monitoring

1. Introduction

PHAs(Potentially harmful applications) are applications (apps) that pose potential security risks and include Trojan, spyware and phishing. Click-fraud apps that click on advertisements without the consent of the user are also considered to be PHAs, and as of 2018, 54.9% of PHAs installed are click-fraud apps. Click-fraud apps appear to perform normal functions like playing music and gaming, but they are misused for click-fraud by executing events that click on advertisements without the user's knowledge [1].

However, while PHAs pose potential risks, they are not always harmful. Therefore, real-time monitoring should be performed for events triggered by PHAs on a user's device. This paper is structured as follows. Section 2 analyzes the

attack scenario of a PHA, and section 3 proposes an event-based PHA detection method. Section 4 presents the conclusions of the research.

2. Analysis of Click-fraud Attack

Cost-Per-Click (CPC) is a method whereby a publisher earns income from an advertiser based on the number of times users clicked on an ad such as a Website banner. Click fraud is an attack exploiting a vulnerability in the CPC revenue structure. Accordingly, this section analyzes a click-fraud attack scenario based on the CPC's revenue structure.

2.1 Revenue structure of CPC

The revenue structure of CPC is as follows. An advertiser registers an ad in the ad network for distribution. Furthermore, a publisher

registers in the ad network and produces an app that uses the ad SDK library to request ads from the ad network and show them to users. Using the corresponding app, the publisher earns revenue from the advertiser based on the number of clicks on the ad [2, 3].

2.2 Click-fraud attack scenario

Fig. 1 is a click-fraud attack scenario in the CPC revenue structure and detail steps are as follows.

Step 1. The advertiser registers an ad in the ad network.

Step 2. The attacker registers as a publisher in the ad network.

Step 3~4. The attacker distributes an app to use for click fraud. If an ad is requested through ad SDK library, attacker distributes the app with ad SDK library. From the user's perspective, the corresponding app performs a normal function such as providing music, a camera feature, or a game.

Step 5. Using the ad SDK library embedded in the click-fraud app, the attacker requests an ad from the ad network without the consent of the user or requests an ad URL directly from the ad network without using the ad SDK library.

Step 6. The attacker receives the revenue from the advertiser corresponding to the click fraud [1, 4].

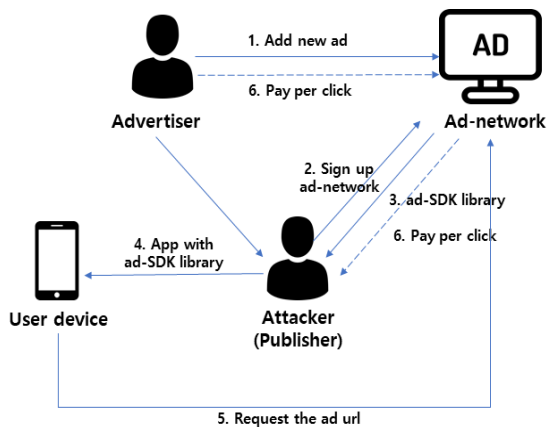


Fig. 1. Click-fraud attack scenario

3. Event-based traffic Monitoring Architecture

Although a click-fraud app appears to be performing a normal function, it sends fraudulent click events in the background without user consent. If risky permissions such as the permission to read phonebook and text messages are given to the click-fraud app, the attacker can perform additional risky tasks such as stealing the user's phonebook, leaking private information, and downloading other PHAs.

The event-based traffic monitoring architecture is shown in **Fig. 2**. Whether traffic is related to an ad or not is determined through several modules to detect a click-fraud app posing such potential risks, which are described in this section. After traffic is determined to be related to an ad, screen touch event information is combined with the traffic information to determine whether or not click fraud occurred.

3.1 Ad Traffic Determination Modules

The traffic collector in **Fig. 2** obtains the traffic in user device. The following Three modules help to determine whether or not collected traffic is related to an ad. By combining the results of modules below, it can be determined whether or not traffic is related to an ad.

• EasyList Ad-blocking Filter

By filtering the URL, DOM element, and third-party ad domain, whether traffic is related to an ad or not is first determined. EasyList is a filter frequently used by ad-blockers, and it determines whether a request is an ad request [5].

• Http Request-Response Relation

Ad request usually has many parameters for identifying the user device, and Ad request usually has images and JavaScript. Therefore, if there are many parameters in the request and an image and JavaScript is in the response, the traffic can be determined to be ad related [6].

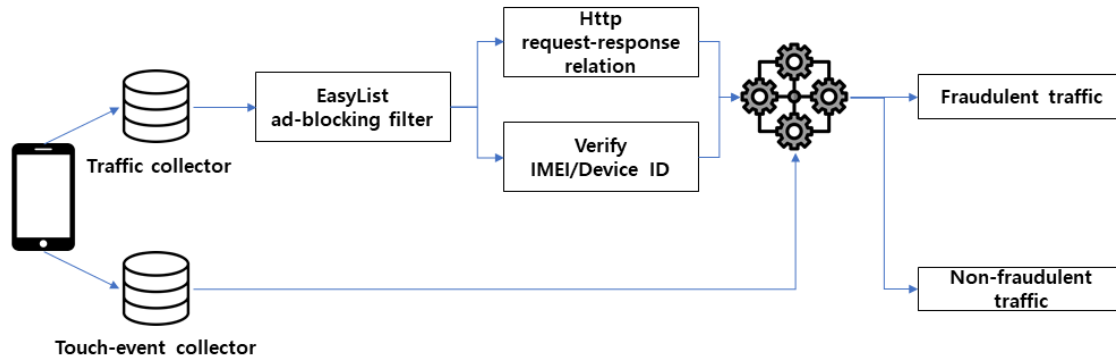


Fig. 2. Event-based Traffic Monitoring Architecture

• IMEI(International mobile equipment identity)/Device ID Modification Check

Ad networks like Caudy Ads place limits on the maximum number of ad requests per IMEI or device ID in a certain period. Attackers modify device IDs, thereby bypassing these limits and generating many ad requests to earn larger ad revenues in a shorter period of time[5]. Therefore, modification check is performed to determine if the device ID has been changed in the settings.db file of a user's device, thereby testing whether an attacker has bypassed it to commit Click-fraud. If the IMEI or device ID has been modified, it can be assumed that Click-fraud has occurred because it indicates that the attacker performed a bypassing task.

3.2 Event-based Click-fraud Traffic Detection

The touch event collector in Fig. 2. obtains the time and location of a touch event using the input_event_from_user function at the kernel level and records them in the touch_events.log file. If an attacker arbitrarily creates an ad-click signal, the touch event is not saved in the touch_events.log file because it is not recorded in the input_event_from_user function.

Fig. 3 is flow chart for determining click fraud combining touch event and ad related traffic in case of background or foreground ad traffic.

• In the case of ad traffic generated in background

If ad traffic occurs in the background without a record of a click event in the touch_events.log file, it can be detected as click-fraud traffic since it is ad traffic generated without the consent of the user.

• In the case of ad traffic generated in foreground

If ad traffic has occurred without a record in the touch_events.log file, it can be detected as Click-fraud traffic since the ad traffic has occurred without the consent of the user.

If an app is in the foreground, however, a touch event and click-fraud traffic can occur simultaneously due to the user is using the device. Therefore, when an app is in the foreground and the touch event and ad traffic have occurred simultaneously, whether ad traffic has occurred at a corresponding coordinate is determined based on the location of the touch event. If the ad traffic has not occurred at the corresponding coordinate, it can be detected as click-fraud traffic since it is ad traffic generated without the consent of the user.

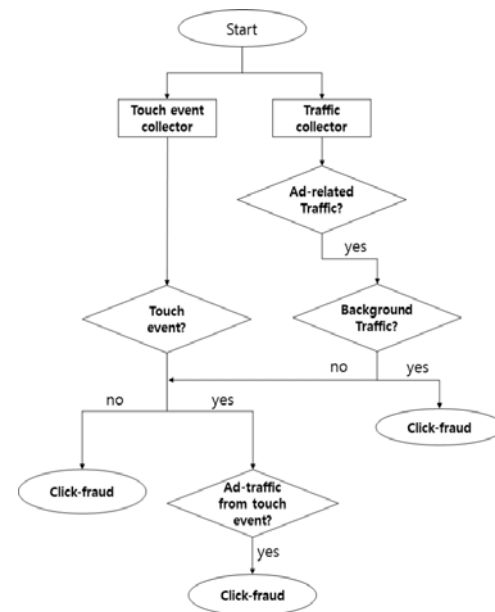


Fig. 3. Flowchart for determining click fraud

4. Conclusions

Click-fraud accounting for half percentage among PHAs sends ad signals without a user's consent through an app. Such a click-fraud app not only sends fraudulent click events but can also perform additional potentially harmful tasks such as leaking private information. Therefore, this paper proposed an event-based traffic monitoring architecture to detect click-fraud apps. Through this architecture, secondary damages and potential risks caused by Click-fraud apps can be reduced, thereby improving security of user.

References

- [1] Android Security & Privacy Team, "Android Security 2018 year in Review", 2019.
- [2] Geumhwan Cho, Junsung ChoM et al., "An empirical study of click fraud in mobile advertising networks" in *International Conference on Availability, Reliability and Security Symposium on Augmented Reality*, 2019.
- [3] Feng Dong, Haoyu Wang et al., "FraudDroid: Automated Ad Fraud Detection for Android Apps" in *Proc. of the 26th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE '18)*, 2018.
- [4] Android Security & Privacy Team, "PHA Family Highlights: Zen and its cousins", 2019.
- [5] AdBlock, <https://help.getadblock.com/support/solutions/articles/6000066909-which-filter-lists-should-i-use->, 2019.
- [6] Md Shahrear Iqbal, Mohammad Zulkernine et al., "FCFraud: Fighting Click-fraud from the User Side" in *2016 IEEE 17th International Symposium on High Assurance Systems Engineering*, 2016.